Attorney's [illegible]t No.: 10559-364001 / P8247-2

APPLICATION

FOR

UNITED STATES LETTERS PATENT

09672345.092800

TITLE: SubA1 ~~ACCESSING MULTI-PORTED MEMORY~~

APPLICANT: DAVID I. POISNER

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL419343051US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

September 28, 2000
Date of Deposit

[illegible]
Signature

Vince Defante
Typed or Printed Name of Person Signing Certificate

Sub A1 > ~~ACCESSING MULTI-PORTED MEMORY~~

This is a continuation in part of Ser. No. 09/572,047, filed May 16, 2000.

BACKGROUND

This invention relates to accessing multi-ported memory.

In a conventional computing system the central processing unit (CPU), main memory and input/output (I/O) devices are connected by a bus. A "bus master" or "bus arbiter" controls and directs data traffic among the components of the computing system.

Main memory is used as the principal site for storing data. An "access" to main memory writes data to or reads data from main memory. Making an access (or "accessing") is typically preceded by a request for access from another component of the system, such as the CPU or an I/O device, followed by a grant of permission by the bus arbiter.

There are two principal types of accesses. The first type is a data access, in which large amounts of data are written to or read from main memory. A data access may be on the order of thousands of bytes. The second type is a control/status access, characterized by a small number of reads or writes to a defined data structure in order to report the status of an input/output device, process data, or initiate some input/output activity. In contrast to data accesses, a control/status access is usually on the order of a few bits. Control accesses are generally initiated by the CPU, while status accesses are generally initiated by the I/O devices.

DESCRIPTION OF DRAWINGS

Figs. 1 and 2 are conceptual block diagrams depicting an embodiment of the invention.

SubA2

Fig. 3 is a conceptual block diagram depicting a component shown in Fig. 2, illustrating an embodiment of the invention.

DETAILED DESCRIPTION

In Fig. 1, a computer architecture 10 includes a central processing unit (CPU) 12, main memory 22 and I/O devices 26. Main memory 22 is generally a form of random access memory (RAM), such as dynamic RAM (DRAM), Rambus™ DRAM (RDRAM), synchronous DRAM (SDRAM) or SyncLink DRAM (SLDRAM). Communications among these components are regulated by a memory controller 16. Memory controller 16 performs the functions of a bus arbiter by managing communications 14 with CPU 12, communications 20 with main memory 22, and communications 24 with I/O devices 26.

Memory controller 16 includes a dual-ported memory 18 that can be accessed by CPU 12 and by I/O device 26 independently. Dual-ported memory 18 can be any form of random access memory although static random access memory, or SRAM, is useful because of its speed and reliability, and some forms of DRAM are useful because DRAM saves chip space. In the embodiments described below, the memory is dual-ported, but the invention is not necessarily limited to two ports. Dual-ported memory 18 may be integrated into a single integrated circuit with memory controller 16, which improves speed and saves chip space.

Typically control/status accesses waste system resources because they are generally short and have comparatively few bits. Many forms of main memory, such as RDRAM, handle long strings of bits more efficiently than short strings of bits. In addition, as will be discussed below, many conventional computer architectures include a cache memory and ensure cache consistency by employing "bus snooping," i.e., monitoring the communication channels for write accesses to main memory. Bus snooping may require additional logic to be present in each microprocessor and, therefore, can increase the cost and decrease the performance of the microprocessor. The performance degradation incurred by bus snooping is accelerated as the number of short, control/status accesses increases relative to the number of data accesses.

Computer architecture 10 offers a more efficient use of resources by directing control/status accesses to dual-ported memory 18 in controller 16, rather than to main memory 22. By directing control/status accesses to dual-ported memory 18 instead of main memory 22, a bus snoop and its attendant delay are avoided.

There are several ways in which accesses from CPU 12 and I/O device 26 may be directed to dual-ported memory 18. One way is for the computer's operating system to direct appropriate accesses to dual-ported memory 18. Another way is for components to be configured to look for dual-ported memory 18. For example, a controller for I/O device 26 may check controller 16 to see whether dual-ported memory 18 is available, and if dual-ported memory 18 is available, I/O device 26 may use dual-ported memory 18. If dual-ported memory 18 is not available, I/O device 26 may use main memory 22. Other ways in which accesses may be directed to controller's memory exist, and these are not exclusive of each other.

Fig. 2 is a block diagram depicting a computer architecture 40 that includes an embodiment of the invention. Fig. 2 is more detailed than Fig. 1, and shows more of the components typically present within a computer architecture than are shown in Fig. 1. Architecture 40 includes main memory 46 electrically coupled to memory controller 50. Architecture 40 further comprises CPU 42, which is electrically coupled to memory controller 50 and to cache memory 44. Cache memory 44 comprises high-speed RAM for holding data and instructions that, for example, are contained in main memory 46, or originate from main memory 46, and that are recently used by CPU 42. For cache memory 44 to be useful, the information stored in cache memory 44 must be consistent with the data and instructions held in main memory 46. As previously mentioned, bus snooping is a technique used to maintain cache consistency. Cache memory 44 can copy data and instructions from main memory 46 via memory controller 50. In addition, memory controller 50 is coupled to I/O bus 54. Other components may be electrically coupled to memory controller 50 in addition to those shown in Fig. 2.

I/O bus 54 is a communication channel typically dedicated to input/output operations. I/O bus 54 may be a high-speed bus supporting peripherals operating at high data throughput rates, such as a Peripheral Component Interconnect (PCI) bus. The term "peripheral" encompasses but is not limited to external devices attached to the computer system. I/O devices are the most common peripherals. A computer system may have many peripherals, and these devices may interface with the system by way of I/O bus 54.

In Fig. 2, three peripherals are shown for exemplary purposes: an external storage device 62 such as a hard disk, a display device 64 such as a monitor, and a printer 66 such as a laser printer. The peripherals 62, 64, 66 interface with the system through I/O controllers 56, 58, 60, respectively. Although three peripherals 62, 64, 66 are shown, I/O bus 54 is not necessarily limited to use by three peripherals. Other peripherals include devices such as a modem, keyboard, optical drive, network connection, and mouse.

Memory controller 50 also provides a communication connection among components such as CPU 42 and main memory 46. For example, memory controller 50 directs data traffic among CPU 42, main memory 46 and I/O bus 54. Bus arbiter 48, included in memory controller 50, regulates data traffic on I/O bus 54. In addition, memory controller 50 includes dual-ported memory 52, independently accessible by CPU 42 or by any peripheral 62, 64, 66 via I/O bus 54. More specifically, the operating system executing on CPU 42 treats dual-port memory 52 as non-cached memory. As a result, read and write accesses to dual-ported memory 52 do not implicate cache memory 44 and need not be monitored. Therefore, dual-ported memory 52 may be accessed by CPU 42 or by any peripheral 62, 64, 66 without a bus snoop, avoiding the delay caused by the bus snoop.

In this architecture 40, control/status accesses are directed principally to dual-ported memory 52, rather than to main memory 46, thereby avoiding a bus snoop and its attendant delay. The advantages described above, such as speed and efficiency, can be attained with this architecture 40, and "smart" addressing can be used to map a virtual address to a physical address. With smart addressing, addresses are mapped to simplify conversion of the virtual address to the physical address. Smart addressing can avoid operating system calls and can save time by relating virtual and physical addresses through a simple mapping algorithm. Byte-aligning addresses of a page boundary on an external disk 62 with addresses in dual-ported memory 52 is one example of smart addressing. With the addresses byte-aligned, the addresses are related by the addition or subtraction of a constant. With such a mapping, it is not necessary for a device driver to call the operating system to convert a virtual address to a physical address; the device driver can translate the virtual address to a physical address simply by adding or subtracting.

As is evident from Fig. 2, increased usage of dual-ported memory 52 by peripherals results in reduced usage of main memory 46 and consequently less data traffic on the channel 45 connecting main memory 46 to memory controller 50. With less traffic on channel 45, CPU 42 can have greater access to main memory 46, and more information can be transferred between CPU 42 and main memory 46 in a given period of time.

Memory controller's dual-ported memory 52 need not be restricted to reading or writing control/status accesses. In some circumstances, data may be stored in memory 52.

As shown in Fig. 3, dual-ported memory 52 can be divided into general-purpose memory 80 and reservation bits memory 82. General-purpose memory 80 may be subdivided into regions or blocks of memory, which may be allocated for use by CPU 42 or by I/O device 62, 64 and 66. Each block may be uniquely identified with a reservation bit, such that for N blocks there are N corresponding reservation bits. For example, the first reservation bit 84 is identified with the first block 86. The number of required reservation bits depends upon the size of general-purpose memory 80 and upon the size of a block. For example, if a block consists of 128 bytes, and if general-purpose memory 80 includes 4 kilobytes of memory, then 32 blocks would be available and 32 reservation bits would be needed (32 bits x 128 bytes/bit = 4 kilobytes).

The state of a reservation bit denotes whether the reservation bit's corresponding memory block in general-purpose memory 80 has been allocated. For example, a '0' value bit in the first reservation bit 84 may signify that the first memory block 86 is free, while a '1' value bit in the first reservation bit 84 may signify that the first memory block 86 has been allocated for use. When a block of general-purpose memory 80 in dual-ported memory 52 is needed, the reservation bits 82 corresponding to the blocks can be checked to determine whether any blocks are free. When a block in general-purpose memory 80 has been allocated but is no longer needed, the block can be de-allocated by clearing the corresponding reservation bit to indicate that the block is free. In the event all reservation bits 82 show that all general-purpose memory 80 has been allocated, then main memory 48 may be accessed instead.

Sub A3

~~Fig. 3 exemplifies one of many possible memory organizations in dual-ported memory 52. Reservation bits 82 need not be grouped together as shown in Fig. 3, but may, for example, be denoted as the initial bit of each block. Furthermore, each block of general-purpose memory may have multiple reservation bits mapped to the block.~~

Other embodiments are within the scope of the following claims.